DDoS Self-Defense

Joe Stewart
Director of Malware Research
SecureWorks Counter Threat Unit
State of DDoS A&D

• Attacks
  – Dozens of attack tools, methods
    • Some more effective than others
    • Readily-available to script kiddies
  – Constantly happening, but not often in the news
  – Typical attack targets:
    • IRC servers
    • Small business
    • Gaming sites
    • Rival botnets
    • Whitehats

• Defenses
  – Commercial solutions available for $$ (₩ ₩ ₩)
  – Few alternative options other than “suffer through it”
DDoS Self-Defense

• Other options are in fact available, just not widely known or used
  – Countermeasures may be in legal grey-area
  – Difficulty in quickly bootstrapping defenses during an attack
  – Difficulty in quickly locating contacts/resources who can assist with defense

• Solution (and purpose of this talk)
  – Review legal issues around network self-defense
  – Understand active and passive network self-defense techniques
  – Find out about whitehat communities and build contacts
Passive Defense: Tarpitting 1

• Many attacks are HTTP-based resource exhaustion attacks
  – Synfloods not always effective against targets since servers/network providers have gotten better at dealing with them
  – Instead of “filling up the pipe”, it’s easier to overload the webserver’s max connections or available CPU/memory resources

• Most HTTP-based attacks launched in a userspace process, therefore:
  – Must use the system TCP/IP stack
  – Are limited by the rules implemented by the TCP/IP stack
  – We can take advantage of this
Passive Defense: Tarpitting 2

- On DDoS victim server:
  - Identify and handle attacker connections
  - Immediately set TCP window size to few or zero bytes
  - Send no more packets, forget about the connection

- On attacker bot machine:
  - Stack must obey the TCP window size setting and sends no more data than will fit in the window before receiving an ACK
  - Since no ACK ever comes, attacker tries to resend request no larger than the window size at ever-increasing intervals forever or until bot kills the connection

- Traffic destined to victim is significantly decreased
Passive Defense: Tarpitting 3

Bot Throughput During DDoS

DDoS Mitigation Bandwidth Comparison

Server Response:  
- Accept  
- Drop  
- Tarpit
Bot CPU Load During DDoS

DROP response

TARPIT response
Passive Defense: Tarpitting 5

• Software for tarpitting
  – LaBrea by Tom Liston
    • No longer distributed by Tom
    • Source code available from other sites
  – Linux Netfilter
    • `iptables -A INPUT -s x.x.x.x -p tcp -j TARPIT`

• Further reading
  – The University of Florida used tarpitting to defend against NetSky worm DDoS attack in 2004:
Browser-Based Attacks

• Often used during “hacktivist” activities
• No botnet required
  – HTML/javascript page distributed to willing attackers
  – Script continually reloads pages/images from victim website
  – Easy to deploy
    • Download and edit HTML page to add targets
    • Hand it out in a forum with simple instructions: “open this in your browser and let it run”

• Example: Lad Vampire
  – originally written to attack phishing pages by anti-phishing-fraud vigilante group
  – No longer distributed by same group, but still in active circulation
Lad Vampire in Action

10lwUt?

6 images, (6 active, 0 faltering, 0 suffering, 0 dead). 1566 loaded (17.7/s), 26 failed. 5.947 MB total (76.72 KB/s)
GET /images/bg.jpg?1264undefined HTTP/1.1
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: joestewart.org
Connection: Keep-Alive
Lad Vampire Mitigation

```
RewriteEngine on
RewriteCond %{QUERY_STRING} ^[0-9]+undefined
RewriteRule /.*/\.(jpg|gif)$ /cgi-bin/log.cgi
```
Warning Image CGI Script

/cgi-bin/log.cgi

#!/usr/bin/perl
use GD::Simple;
my $i = GD::Simple->new(130, 90);
$i->bgcolor('red'); $i->fgcolor('black');
$i->rectangle(1,1,129,89); $i->moveTo(20,30);
$i->string('Your IP address'); $i->moveTo(25,40);
$i->string($ENV{'REMOTE_ADDR'});$i->moveTo(20,50);
$i->string('has been logged'); $i->moveTo(25,50);
$i->string('and will be reported'); $i->moveTo(20,60);
$i->string('to the authorities.');
print "Content-Type: image/png\n\n";
print $i->png;
Lad Vampire Mitigation Result

6 images, (6 active, 0 faltering, 0 suffering, 0 dead). 1290 loaded (4.2/s), 26 failed. 3.439 MB total (8.589 KB/s)

Your IP address xxx.xxx.xxx.xxx has been logged and will be reported to the authorities.
joestewart.org
S:61 F:0

Your IP address xxx.xxx.xxx.xxx has been logged and will be reported to the authorities.
joestewart.org
S:56 F:0

Your IP address xxx.xxx.xxx.xxx has been logged and will be reported to the authorities.
joestewart.org
S:58 F:0

Your IP address xxx.xxx.xxx.xxx has been logged and will be reported to the authorities.
joestewart.org
S:57 F:0

Your IP address xxx.xxx.xxx.xxx has been logged and will be reported to the authorities.
joestewart.org
S:56 F:0

Your IP address xxx.xxx.xxx.xxx has been logged and will be reported to the authorities.
joestewart.org
S:48 F:0
Active Defense: Traceback

• With cooperation, it is possible to locate and take down (or take over) control servers for DDoS malware

• Need to establish contact with helpful persons in different business sectors:
  – ISPs
    • Knowing the target IP and network traffic type, ISPs can find infected customers and use network flow triangulation to find the common control server IP
    • Some security monitoring companies have access to similar data
  – Antivirus researchers
    • Knowing the fingerprint of the attack software may enable them to find the actual malware sample that is being used in the attack
    • Replaying the sample in a sandnet can reveal the control server IP
Active Defense: Takedown or Takeover?

• We have the IP of the controller, now what?
• Takedown may not be desirable
  – Losing connectivity with the controller may not cause the bots to stop attacking, in fact it could prolong an attack
  – Depending on the bot, a backup hostname could be in use, so the attacker is back up and running in minutes
  – Finding all backup names and IP addresses involved is crucial
• Takeover
  – Many bot types have no way to authenticate the controller
    • As long as it speaks the right protocol, the bots will obey
  – With cooperation from DNS or hosting provider, bots can be instructed to stop the attack before the final takedown
Active Defense: Becoming the Controller

- **Black Energy**
  - Very popular DDoS bot
  - No authentication of controller
  - Stop command:
    
    `10;2000;10;0;0;30;100;40;20;1000;2000#stop#1#xCOMP_ABCD1234`

- **Illusion Bot**
  - Somewhat less popular DDoS bot
  - No authentication of controller
  - Stop command: `100 @stopall`
GET / HTTP/1.1
Accept: */*
Accept-language: en-us
User-agent: Opera/9.02 (Windows NT 5.1; U; ru)
Host: example.com
Connection: Keep-Alive
Know Your Attacker: Illusion Bot

GET http://example.com//~/~/~/~/~/ HTTP/1.1
Host: example.com
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)
Refer: http://example.com/cgi-bin/index.pl

GET http://example.com/1.php HTTP/1.1
Host: example.com
Accept: */*
User-Agent: Microsoft-WebDAV-MiniRedir/5.1.2600
Refer: http://example.com/index.html

Referrer header is spelled the wrong “wrong” way
GET / HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, application/x-ms-application, application/x-ms-xbap, application/vnd.ms-xpsdocument, application/xaml+xml, */*
Accept-Language: ko
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: %s
Host: [target hostname]
Connection: Keep-Alive

User-Agent selected at random...
<table>
<thead>
<tr>
<th>User-Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; GTB6; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)</td>
</tr>
<tr>
<td>Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; GTB6; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)</td>
</tr>
<tr>
<td>Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; InfoPath.2; MAXTHON 2.0)</td>
</tr>
<tr>
<td>Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; GTB6; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)</td>
</tr>
<tr>
<td>Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.20) Gecko/20081217 Firefox/2.0.0.20 (.NET CLR 3.5.30729)</td>
</tr>
</tbody>
</table>

Firefox UA, but MSIE header-ordering
Active Defense: Counter-Attack 1

- **Safe haven hosting providers**
  - Do not care about attacks controlled from their site
  - Will not respond or cooperate with takedowns
  - Worse, they may share your correspondence with the attacker

- **What are the remedies?**
  - Launch a counter-attack against the DDoS control server
    - Probably not legal most places
    - May prolong the attack same as with takedown
  - Work with ISP/Security community
    - Null route the controller IP or netblock from the rest of the world
    - Expose uncooperative hosting providers in the press - what’s known as a “Krebsing” (see McColo, 3FN)
Active Defense: Counter-Attack 2

- Many control servers are poorly programmed
- Vulnerable to SQL injection
  - Expose admin authentication credentials
  - Enumerate bots
  - Insert commands
- Vulnerable to cross-site scripting attacks (XSS)
  - Add an iframe to the attacker’s stats page and track his IP
    - Add proxy-decloaking code to the iframe for extra credit
- Poorly-thought-out interface with links to third-party sites
  - Reveal the control panel URL in the referrer log
- Legality of taking advantage of these techniques still an issue
/* this function will be used by bots */
if ($act == "online")
{
    if (isset( $_GET["nickname"] ))
        $nickname = base64_decode( $_GET["nickname"] );
    else
        exit();

    if (isset( $_GET["s4"] )) $s4 = $_GET["s4"];
    else $s4 = 0;
    if (isset( $_GET["s5"] )) $s5 = $_GET["s5"];
    else $s5 = 0;
    if (isset( $_POST["msg_out"] )) $msg_out =
        base64_decode( $_POST["msg_out"] );
    else $msg_out = "";
    if (isset( $_GET["msg_out"] )) $msg_out = $_GET["msg_out"];
    else $msg_out = "";
    die(db_bot_online( $nickname, $msg_out, $s4, $s5 ));
/* add/update DB record about bot */
function db_bot_online( $nickname, $mo, $socks4_port, $socks5_port ){
    ...
    $msg_out = str_replace( "\", "\\", $mo );
    $msg_out = htmlspecialchars( $msg_out );
    ...
    else {
        if ($msg_in) $st = 0; else $st = 1;
        mysql_query( "INSERT INTO $mysql_bots_table
VALUES($time, "$ip", "$nickname", "$msg_in", ",", $st, $socks4_port, $socks5_port)" );
    }
}
Illusion Bot XSS Example

/* Online bots listing */
function db_list_bots()
{
    ...
    $nickname = cutstr( $arr["nickname"], 12 );
    $fullnickname = $arr["nickname"];
    ...
    if ($status)
        $nickref = "<td class="tableitem"><a title="Add nick" href="javascript:addn('$fullnickname');">$nickname</a><td>";

nickname is varchar(64), enough room for an iframe tag
Final Word

• Personal networking and information sharing is key
  – Ahead of time, not after-the-fact
• Cybercrime laws not up-to-date
  – Law enforcement is unable to respond in a timely fashion to protect the innocent
  – Fear of prosecution keeps those who are able to respond from doing so with all measures available
  – Untested, but some jurisdictions have “nuisance laws” that protect citizens who take action that would be otherwise illegal
  – Legislators need to understand the issues involved and provide options for self-defense without fear of incarceration
Questions?